

before interruption. This accelerates the process of handling a nested fault. Having two copies of a resource also eliminates the need to have N copies of a resource to support N levels of nesting.

The nested fault handler also includes (or cooperates with) an epilogue portion which is executed after the nested fault handler to restore the values in the error program control register and the fault handling registers before returning to the first level fault handler from the nested fault handler. This epilogue code assures that the first level fault handler begins at the correct address and that the state of the first level fault handler is correct before its execution continues.

In order to handle nested faults within nested faults, state relating to each level of nested fault is also stored and restored when the fault handler has completed. This may be done by duplicating hardware such as the registers 10, 12, and 14 or by storing the state in memory (e.g., cache memory which may be rapidly accessed). In any case, the method proceeds in the manner described for the nested fault process in either Figure 2 or Figure 3.

Although the present invention has been described in terms of a preferred embodiment, it will be appreciated that various modifications and alterations might be made by those skilled in the art without departing from the spirit and scope of the invention. The invention should therefore be measured in terms of the claims which follow.

What Is Claimed Is:

1 Claim 1. A method for resolving nested faults including the steps of
2 determining whether a fault is a first level fault, responding to a
3 determination of a first level fault by saving a first amount of state
4 sufficient to handle a first level fault, and responding to a determination
5 of a nested fault by saving an additional amount of state before handling
6 the fault.

1 Claim 2. A method as claimed in Claim 1 in which the step of
2 determining whether a fault is a first level fault comprises storing an
3 indication when a first level fault handler is executed, and determining if
4 an indication has been stored when a level fault handler is executed.

1 Claim 3. A method as claimed in Claim 1 further including the steps
2 of executing a first level fault handler after saving a first amount of state
3 sufficient to handle a first level fault, and executing a different fault
4 handler after saving an additional amount of state before handling the
5 fault.

1 Claim 4. A method as claimed in Claim 3 further including the steps
2 of executing epilogue sequences of instructions after handling any fault
3 handler enabling return of execution to a correct sequence of
4 instructions following a fault last handled.

1 Claim 5. A fault handler system comprising means for determining
2 whether a fault is a first level fault, means responding to a determination
3 of a first level fault by saving a first amount of state sufficient to handle a
4 first level fault, and means for responding to a determination of a nested
5 fault by saving an additional amount of state before handling the fault.

1 Claim 6. A processor fault handler as claimed in Claim 5 in which the
2 means for determining whether a fault is a first level fault includes
3 storage media for holding an indication that a processor is handling a
4 fault, and means for detecting the condition of the storage media.

1 Claim 7. A processor fault handler as claimed in Claim 5 in which the
2 storage media is a processor register.

1 Claim 8. A processor fault handler as claimed in Claim 5 in which the
2 means for detecting the condition of the storage media includes software
3 running on the processor.

1 Claim 9. A processor fault handler as claimed in Claim 5 in which the
2 means for responding to a determination of a nested fault by saving an
3 additional amount of state before handling the fault includes fault
4 handling registers in addition to a first set of fault handling registers.

Ad
C